Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 04-022

Soft Clustering Criterion Functions for Partitional Document Clustering

Ying Zhao and George Karypis

May 26, 2004

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
26 MAY 2004		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Soft Clustering Criterion Functions for Partitional Document Clustering	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Army Research Laboratory,2800 Powder Mill Road,Adelphi,MD,20783-1197**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**12**	

Soft Clustering Criterion Functions for Partitional Document Clustering *

Ying Zhao
University of Minnesota, Department of Computer Science and Engineering
Minneapolis, MN 55455
yzhao@cs.umn.edu

George Karypis
University of Minnesota, Department of Computer Science and Engineering, Digital Technology Center, and Army HPC Research Center, Minneapolis, MN 55455
karypis@cs.umn.edu

ABSTRACT

Recently published studies have shown that partitional clustering algorithms that optimize certain criterion functions, which measure key aspects of inter- and intra-cluster similarity, are very effective in producing hard clustering solutions for document datasets and outperform traditional partitional and agglomerative algorithms. In this paper we study the extent to which these criterion functions can be modified to include soft membership functions and whether or not the resulting soft clustering algorithms can further improve the clustering solutions. Specifically, we focus on four of these hard criterion functions, derive their soft-clustering extensions, present a comprehensive experimental evaluation involving twelve different datasets, and analyze their overall characteristics. Our results show that introducing softness into the criterion functions tends to lead to better clustering results for most datasets and consistently improve the separation between the clusters.

Keywords

Document clustering, Soft clustering

1. INTRODUCTION

Fast and high-quality document clustering algorithms play an important role in helping users to effectively navigate, summarize, and organize an enormous amount of text documents available on the Internet, digital libraries, news sources, and company-wide intranets. Over the years a variety of different algorithms have been developed. These algorithms can be categorized along different dimensions based either on the underlying methodology of the algorithm, leading to *agglomerative* [36, 24, 15, 16, 22] or *partitional* approaches [28, 20, 32, 8, 41, 19, 38, 7, 13], or on the nature of the membership function, leading to *hard (crisp)* or *soft (fuzzy)* solutions.

In recent years, soft clustering algorithms have been studied in document clustering and shown to be effective [29, 25, 30] in finding both overlapping and non-overlappying clusters. Studies have shown that "hardening" the results obtained by fuzzy C-means produces better hard clustering solutions than direct K-means [17], which suggests that including soft membership functions into other criterion functions may lead to better hard clustering solutions as well.

Recently, we studied seven different hard partitional clustering criterion functions in the context of document clustering, which optimize various aspects of intra-cluster similarity, inter-cluster dissimilarity, and their combinations [45, 44, 46]. Our experiments showed that different criterion functions lead to substantially different results, whereas our analysis showed that their performance depends on the degree to which they can correctly operate when the dataset contains clusters of different densities (*i.e.*, they contain documents whose pairwise similarities are different) and the degree to which they can produce balanced clusters. We also showed that among these seven criterion functions, there are a set of criterion functions that consistently outperform the rest.

The focus of this paper is to extend four of these hard criterion functions ($\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{E}_1$, $\mathcal{G}_1$ [45]) to allow soft membership functions, and to see whether or not introducing softness into these criterion functions leads to better clustering solutions. These criterion functions were selected because they include some of the best- and worst-performing schemes, and represent some of the most widely-used criterion functions for document clustering. We developed a hard-clustering based optimization algorithm that optimizes the various soft criterion functions. Since this optimization algorithm simultaneously produces both a hard and a soft clustering solution, we focused on evaluating the hard clustering solution and compared it with the one obtained by the hard criterion functions. We present a comprehensive experimental evaluation involving twelve different datasets. Our experimental results show that introducing softness into the criterion functions tends to consistently improve the separation between the clusters. Although the experimental results show some dataset dependency, for most datasets the soft criterion

*This work was supported in part by NSF CCR-9972519, EIA-9986042, ACI-9982274, ACI-0133464, and ACI-0312828; the Digital Technology Center at the University of Minnesota; and by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under Cooperative Agreement number DAAD19-01-2-0014. The content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

functions tend to lead to better clustering results. Moreover, our experimental results show that the soft clustering extension of the worst-performing hard criterion function ($\mathcal{I}_1$) achieves the best relative improvement.

The rest of this paper is organized as follows. Section 2 provides some information on how documents are represented and how the similarity or distance between documents is computed. Section 3 discusses some existing soft clustering algorithms related to our work. Sections 4 and 5 describe the four hard criterion functions that are the focus of this paper and presents their soft clustering extensions, respectively. Section 6 describes the algorithm that optimizes the various soft criterion functions and the clustering algorithm itself. Section 7 provides the detailed experimental evaluation of the various soft criterion functions. Section 8 discusses some important observations from the experimental results. Finally, Section 9 provides some concluding remarks and future research directions.

2. PRELIMINARIES

Through-out this paper we will use the symbols n, m, and k to denote the number of documents, the number of terms, and the number of clusters, respectively. We will use the symbol S to denote the set of n documents that we want to cluster, $S_1, S_2, \ldots, S_k$ to denote each one of the k clusters, and $n_1, n_2, \ldots, n_k$ to denote the sizes of the corresponding clusters.

We represent the documents using the vector-space model [35]. In this model, each document d is considered to be a vector in the space of the distinct terms present in the collection. We employ the *tf-idf* term-weighting scheme that represents each document d as the vector

$$d_{tfidf} = (tf_1 \times idf_1, tf_2 \times idf_2, \ldots, tf_m \times idf_m).$$

In this scheme, tf_i corresponds to the frequency of the ith term in the document and $idf_i = \log(n/df_i)$ corresponds to its inverse document frequency in the collection (df_i is the number of documents that contain the ith term). To account for documents of different lengths, we scale the length of each document vector so that it is of unit length.

We measure the similarity between a pair of documents d_i and d_j by taking the cosine of the angle formed between the *tf-idf* representation of their vectors. Specifically, this is defined as

$$\cos(d_i, d_j) = \frac{{d_i}^t d_j}{\|d_i\|\|d_j\|},$$

which can be simplified to $\cos(d_i, d_j) = {d_i}^t d_j$, since the document vectors are of unit length. This similarity measure becomes one if the document vectors point to the same direction (*i.e.*, they contain identical set of terms in the same relative proportion), and zero if there is nothing in common between them (*i.e.*, the vectors are orthogonal to each other).

Finally, given a set A of documents and their corresponding vector representations, we define the ***composite*** vector D_A to be $D_A = \sum_{d \in A} d$, and the ***centroid*** vector C_A to be $C_A = D_A/|A|$.

3. RELATED RESEARCH

Soft clustering that allows an object to appear in multiple clusters has been studied extensively and still remains of great interest. As many datasets and application domains require soft clustering solutions. The fuzzy C-means algorithm [3] is one of the most widely used soft clustering algorithms. It is a soft version of the K-means algorithm that uses a soft membership function. Given a set of objects $d_1, d_2, ..., d_m$, the fuzzy C-means algorithm tries to optimize a least-squared error criterion:

$$J_m = \sum_{r=1}^{k} \left(\sum_{i=1}^{N} \mu_{i,r}^m \|d_i - \overline{C}_r\|^2 \right),$$

where $\mu_{i,r}$ is the d_i's membership in the rth fuzzy cluster satisfying $\sum_{r=1}^{k} \mu_{i,r} = 1, \forall i$, m is the fuzzy factor, and $\overline{C}_r$ is the fuzzy centroid. This minimization problem can be solved analytically by using Lagrange multipliers, and the optimization can be achieved by iteratively updating the membership function and fuzzy centroids as follows:

$$\mu_{i,j} = \frac{(1/\|d_i - \overline{C}_j\|^2)^{1/(m-1)}}{\sum_{r=1}^{k}(1/\|d_i - \overline{C}_r\|^2)^{1/(m-1)}}$$

$$\overline{C}_r = \frac{\sum_{i=1}^{N} \mu_{i,r}^m d_i}{\sum_{i=1}^{N} \mu_{i,r}^m}.$$

The fuzzy factor m controls the fuzzyness of the clustering solution. In general, the fuzzyness of the clustering solution increases as the value of m increases and vice versa. As m approaches one, the algorithm behaves more like standard K-means.

Other newly developed soft clustering algorithms differ from fuzzy C-means by employing different dissimilarity functions [4, 29], or by including both a soft membership function and a weight function (measuring the contribution of each object in a fuzzy cluster) in the criterion functions (robust fuzzy C-means [21] and K-harmonic means [43]). Hamerly and Elkan [17] provided an interesting comparison between K-means, fuzzy C-means, K-harmonic means and two other variates to show the effectiveness of various soft membership functions and weight functions.

Soft clustering has been applied to document clustering and shown to be effective [29, 25, 30]. One of the limitations of classic fuzzy C-means in document clustering is the use of Euclidean distance. Hence, the focus of that research has been on exploring similarity measures that are more suitable for document clustering, for example, cosine similarity (Mendes et al. [29]). To our knowledge, extending other effective criterion functions (for example, $\mathcal{E}_1$ and $\mathcal{G}_1$) with soft membership functions for document clustering has not been studied in the literature.

Another approach to soft clustering proposed by Backer is induced fuzzy partitioning [1]. The key idea is that a hard clustering solution is always mentained in the optimization process. The optimization process starts from an initial hard parition and consists of a number of iterations. During each iteration, the soft membership function is estimated based on the affinity that each object has for hard clusters to induce a soft partition. Then, the hard partition is modified in a way such that the new induced soft partition leads to a better criterion function value. The optimization stops when no modification of the hard partition can be made. Notice that after the optimization process terminates, there is a pair of clustering solutions: a hard clustering solution and the induced soft one. Our proposed optimization algorithm is similar to induced fuzzy partitioning and we focus on evaluating the hard clustering solution obtained by this optimization.

4. HARD CLUSTERING CRITERION FUNCTIONS

A key characteristic of many partitional clustering algorithms is that they use a global criterion function whose optimization drives the entire clustering process. For some of these algorithms the criterion function is implicit (*e.g.*, PDDP [7]), whereas for other algorithms (*e.g*, K-means [28] and Autoclass [8, 10]) the criterion function is explicit and can be easily stated. This later class of algorithms can be thought of as consisting of two key components. First is the criterion function that needs to be optimized by the clustering solution, and second is the actual algorithm that achieves this optimization.

Recently, we studied seven different hard partitional clustering criterion functions in the context of document clustering, which optimize various aspects of intra-cluster similarity, inter-cluster dissimilarity, and their combinations [45, 44, 46]. Our experiments showed that different criterion functions lead to substantially different results, whereas our analysis showed that their performance depends on the degree to which they can correctly operate when the dataset contains clusters of different densities (*i.e.*, they contain documents whose pairwise similarities are different) and the degree to which they can produce balanced clusters.

In this paper, due to space constraints, we focus on only four out of these seven criterion functions, which are referred to as $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{E}_1$, and $\mathcal{G}_1$ [45, 46]. This subset represents some of the most widely-used criterion functions for document clustering, and includes some of the best- and worst-performing schemes. A short description of these functions is presented in the rest of this section, and the reader should consult [45] for a complete description and motivation.

The $\mathcal{I}_1$ criterion function (Equation 1) maximizes the sum of the average pairwise similarities (as measured by the cosine function) between the documents assigned to each cluster weighted according to the size of each cluster and has been used successfully for clustering document datasets [34].

$$\text{maximize } \mathcal{I}_1 = \sum_{r=1}^{k} n_r \left(\frac{1}{n_r^2} \sum_{d_i, d_j \in S_r} \cos(d_i, d_j) \right). \quad (1)$$

The $\mathcal{I}_2$ criterion function (Equation 2) is used by the popular vector-space variant of the K-means algorithm (also referred to as *spherical K-means*) [9, 26, 12, 37]. In this algorithm each cluster is represented by its centroid vector and the goal is to find the solution that maximizes the similarity between each document and the centroid of the cluster that is assigned to.

$$\text{maximize } \mathcal{I}_2 = \sum_{r=1}^{k} \sum_{d_i \in S_r} \cos(d_i, C_r). \quad (2)$$

The $\mathcal{E}_1$ criterion function (Equation 3) computes the clustering by finding a solution that separates the documents of each cluster from the entire collection. Specifically, it tries to minimize the cosine between the centroid vector of each cluster and the centroid vector of the entire collection. The contribution of each cluster is weighted proportionally to its size so that larger clusters will be weighted higher in the overall clustering solution.

$$\text{minimize } \mathcal{E}_1 = \sum_{r=1}^{k} n_r \cos(C_r, C). \quad (3)$$

The $\mathcal{G}_1$ criterion function (Equation 4) is derived by modeling the relationships between the documents using the document-to-document similarity graph G_s [2, 42, 11]. G_s is obtained by treating the pairwise similarity matrix of the dataset as the adjacency matrix of G_s. The $\mathcal{G}_1$ function [13] views the clustering process as that of partitioning the documents into groups that minimize the edge-cut of each partition. However, because this edge-cut-based criterion function may have trivial solutions the edge-cut of each cluster is scaled by the sum of the cluster's internal edges [13]. Since the similarity between each pair of documents is measured using the cosine function, the edge-cut between the rth cluster and the rest of the documents (*i.e.*, $\text{cut}(S_r, S - S_r)$ and the sum of the internal edges between the documents of the rth cluster are given by the numerator and denominator of Equation 4, respectively.

$$\text{minimize } \mathcal{G}_1 = \sum_{r=1}^{k} \frac{\sum_{d_i \in S_r, d_j \in S - S_r} \cos(d_i, d_j)}{\sum_{d_i, d_j \in S_r} \cos(d_i, d_j)}. \quad (4)$$

5. SOFT CLUSTERING CRITERION FUNCTIONS

A natural and straight-forward way of deriving *soft* clustering solutions is to assign each document to multiple clusters. This is usually achieved by using membership functions [29, 17, 3, 1] that for each document d_i and cluster S_j, they compute a non-negative weight, denoted by $m_{i,j}$, such that $\sum_j m_{i,j} = 1$, which indicates the extent to which document d_i belongs to cluster S_j. Thus, we can think of the various $m_{i,j}$ values as the *fraction* by which d_i belongs to cluster S_j. Note that in the case of a hard clustering solution, for each document d_i one of these $m_{i,j}$ values is one (the one corresponding to the cluster that d_i belongs to) and the rest will be zero.

Using these membership functions, the soft clustering extensions of the hard criterion functions described in Section 4 can be derived as follows.

Soft $\mathcal{I}_1$ Criterion Function. Since each cluster can now contain fractions of all the documents, a natural way of measuring the overall pairwise similarity between the documents assigned to each cluster is to take into account their membership functions. Specifically, we can compute the pairwise similarity between the (fractional) documents assigned to the rth soft cluster as

$$\sum_{i,j} \mu_{i,r} \mu_{j,r} \cos(d_i, d_j).$$

Similarly, we can compute the *soft* size $\overline{n}_r$ of the rth soft cluster as $\overline{n}_r = \sum_i m_{i,r}$. Using these definitions, then the soft $\mathcal{I}_1$ criterion function, denoted by $\mathcal{SI}_1$, is defined as follows:

$$\text{maximize } \mathcal{SI}_1 = \sum_{r=1}^{k} \overline{n}_r \left(\frac{1}{\overline{n}_r^2} \sum_{i,j} \mu_{i,r} \mu_{j,r} \cos(d_i, d_j) \right). \quad (5)$$

Soft $\mathcal{I}_2$ Criterion Function. A soft version of the $\mathcal{I}_2$ criterion function can be obtained by extending the notion of the cluster's centroid vector to soft clusters. Since each soft cluster contains

fractions of documents, its centroid vector should also be calculated based on the fractional documents that it contains. Specifically, we can define the *soft* centroid vector of the rth soft cluster $\overline{C}_r$ as

$$\overline{C}_r = \frac{\sum_{i=1}^{N} \mu_{i,r} d_i}{\overline{n}_r},$$

which takes into account the fractional membership of each document and its soft size. Using the above definition, the soft $\mathcal{I}_2$ criterion function, denoted by $\mathcal{SI}_2$, can be obtained by requiring the clustering solution to maximize the similarity between the (fractional) documents assigned to a soft cluster and its centroid. This is formally defined as follows:

$$\text{maximize } \mathcal{SI}_2 = \sum_{r=1}^{k} \left(\sum_{i=1}^{N} \mu_{i,r} \cos(d_i, \overline{C}_r) \right). \quad (6)$$

Soft $\mathcal{E}_1$ Criterion Function. The $\mathcal{E}_1$ criterion function tries to separate the centroid of each cluster from that of the entire collection and weights each cluster by its size. Thus, the new element being introduced in trying to develop a soft version of the $\mathcal{E}_1$ criterion function (over those introduced by $\mathcal{I}_1$ and $\mathcal{I}_2$) is the notion of the collection centroid. In our soft formulation of $\mathcal{E}_1$, we compute this centroid by treating the entire collection as one soft cluster. In this case, the value of the membership function for each document to this cluster is one, and as a result, the soft collection centroid is the same as that for hard clustering; that is, $\overline{C} = C = \sum_{i=1}^{N} d_i/N$). Given this definition and the earlier definitions of soft cluster centroid and soft cluster size, the soft $\mathcal{E}_1$ criterion function, denoted by $\mathcal{SG}_1$, is defined as follows:

$$\text{minimize } \mathcal{SE}_1 = \sum_{r=1}^{k} \overline{n}_r \cos(\overline{C}_r, C). \quad (7)$$

Soft $\mathcal{G}_1$ Criterion Function. In order to develop a soft version of the $\mathcal{G}_1$ criterion function we need to properly define (i) the edge-cut between a cluster and the rest of the documents in the collection, and (ii) the sum of the weights of the edges between the documents in each cluster. Since the weights of the edges between each pair of documents corresponds to the cosine similarity between their respective documents vectors, both of the above quantities can be easily obtained by extending the expressions in Equation 4 to take into account the membership functions. Specifically, we can compute the soft version of the sum of the weights of the edges between the documents of the rth cluster (the denominator of Equation 4) as $\sum_{i,j} \mu_{i,r}\mu_{j,r} \cos(d_i, d_j)$. Similarly, since the rth cluster contains fractions of all the documents, the fractions of the documents that are not assigned to this cluster are the fractions that belong to the cluster corresponding to the "rest" of the documents. As a result, we can compute the edge-cut between the rth cluster and the rest of the documents in the collection as $\sum_{i,j} \mu_{i,r}(1 - \mu_{j,r}) \cos(d_i, d_j)$. Using these definitions, the soft version of the $\mathcal{G}_1$ criterion function, denoted by $\mathcal{SG}_1$, is defined as follows:

$$\text{minimize } \mathcal{SG}_1 = \sum_{r=1}^{k} \frac{\sum_{i,j} \mu_{i,r}(1 - \mu_{j,r}) \cos(d_i, d_j)}{\sum_{i,j} \mu_{i,r}\mu_{j,r} \cos(d_i, d_j)}. \quad (8)$$

6. SOFT PARTITIONAL CLUSTERING ALGORITHM

Our focus thus far has been on developing soft-clustering extensions for four different criterion functions that are used to obtain hard-clustering solutions. We now turn our attention on developing algorithms that compute clustering solutions that optimize each of these criterion functions.

Traditionally, soft clustering algorithms are derived by analytically optimizing their respective criterion functions using Lagrange multipliers (*e.g.*, fuzzy C-means algorithm [3]). This analytical approach leads directly to an iterative strategy that determines the values of the various membership functions by which the overall criterion function is optimized. Even though this approach can be applied to optimize the $\mathcal{SI}_2$ criterion function [29], analytically deriving such optimization iterations for the $\mathcal{SI}_1$, $\mathcal{SE}_1$, and $\mathcal{SG}_1$ functions is hard if not impossible. For this reason, we developed a soft partitional clustering algorithm that determines the values of the membership functions of the various documents following the induced fuzzy partitioning approach [1], and optimizes the soft criterion functions using a hard-clustering based optimization approach.

6.1 Determining the Membership Functions

Given a hard k-way clustering solution $\{S_1, S_2, ..., S_k\}$, we define the membership of document d_i to cluster S_j to be

$$\mu_{i,j} = \frac{\cos(d_i, C_j)^m}{\sum_{r=1}^{k} \cos(d_i, C_r)^m}, \quad (9)$$

where C_r is the centroid of the hard cluster S_r.

The parameter m in Equation 9 is the *fuzzy factor* and controls the "softness" of the membership function and hence the "softness" of the clustering solution (the inclusion of the fuzzy factor was motivated by the formulation of the fuzzy C-means algorithm). When m is equal to zero, the membership values of a document to each cluster are the same (*i.e.*, there is no preference to a particular cluster). On the other hand, as m approaches infinity, the soft membership function becomes the hard membership function (*i.e.*, $\mu_{i,j} = 1$, if d_i is most close to S_j; $\mu_{i,j} = 0$, otherwise). In general, the softness of the clustering solution increases as the value of m decreases and vice versa.

6.2 Determining the Clusters

As we mentioned in Section 3, a hard-clustering based optimization approach results in a pair of clustering solutions: a hard clustering solution and the induced soft clustering solution. In this paper, we focus on the hard clustering solution and used a clustering approach that determines the overall k-way clustering solution by performing a sequence of cluster bisections. In this approach, a k-way solution is obtained by first bisecting the entire collection. Then, one of the two clusters is selected and it is further bisected, leading to a total of three clusters. The process of selecting and bisecting a particular cluster continues until k clusters are obtained. This repeated bisectioning approach was motivated for two reasons. First, recent studies on hard partitional clustering [37, 46] have shown that such an approach leads to better clustering solutions than the traditional approach that computes the k-way solution directly. Second, it leads to an algorithm that has a lower computational complexity (in most cases it is faster by an order of k).

Each of these bisections is performed in two steps. During the first step, an initial clustering solution is obtained by randomly assigning the documents to two clusters. During the second step, the initial clustering is repeatedly refined so that it optimizes the desired clustering criterion function.

The refinement strategy consists of a number of iterations. During

each iteration, the documents are visited in a random order. For each document, d, we compute the change in the value of the soft criterion function obtained by moving d to the other cluster. This is done by deriving the membership values for the original and modified hard clustering solution (*i.e.*, assuming that d moved to the other cluster) and then calculate the change of the soft criterion function. If the change improves the criterion function, then d is moved to the cluster. If no such cluster exists, d remains in the cluster that it already belongs to. The refinement phase ends, as soon as we perform an iteration in which no documents moved between clusters. The detailed pseudo-code and algorithm description refer to Algorithm 6.1.

Algorithm 6.1: SOFT2WAYREFINE(S_1, S_2)

$C_1 \leftarrow$ centroid of S_1
$C_2 \leftarrow$ centroid of S_2
$\mu_{i,j} \leftarrow$ membership values using C_1 and C_2
$\mathcal{F} \leftarrow$ fuzzy criterion function value

while movements are made
  do
    for each $d \in S_1 \cup S_2$
      do
        $(S'_1, S'_2) \leftarrow$ 2-way clustering after moving d
        $C'_1 \leftarrow$ centroid of S'_1
        $C'_2 \leftarrow$ centroid of S'_2
        $\mu'_{i,j} \leftarrow$ membership values using C'_1 and C'_2
        $\mathcal{F}' \leftarrow$ fuzzy criterion function value
        if $\mathcal{F}'$ is better than $\mathcal{F}$
          then
            $S_1 \leftarrow S'_1$
            $S_2 \leftarrow S'_2$
            $\mathcal{F} \leftarrow \mathcal{F}'$
            $\mu_{i,j} \leftarrow \mu'_{i,j}, \quad \forall i, j$

return (S_1, S_2)

Note that unlike the traditional refinement approach used by K-means type of algorithms, the above algorithm moves a document as soon as it is determined that it will lead to an improvement in the value of the criterion function. This type of refinement algorithms are often called *incremental* [14]. Since each move directly optimizes the particular criterion function, this refinement strategy always converges to a local minima.

The greedy nature of the refinement algorithm does not guarantee that it will converge to a global minima, and the local minima solution it obtains depends on the particular set of seed documents that were selected during the initial clustering. To eliminate some of this sensitivity, the overall process is repeated a number of times. That is, we compute N different clustering solutions (*i.e.*, initial clustering followed by cluster refinement), and the one that achieves the best value for the particular criterion function is kept. In all of our experiments, we used $N = 10$. For the rest of this discussion when we refer to the clustering solution we will mean the solution that was obtained by selecting the best out of these N potentially different solutions.

6.2.1 Cluster Selection

A key step in this repeated bisection algorithm is the method used to select which cluster to bisect next. In our experiments, we used a simple strategy of bisecting the largest cluster available at that point of the clustering solution. Our earlier experience with this approach showed that it leads to reasonably good and balanced clustering solutions [37, 45].

We also used a strategy to stop bisecting a cluster. Specifically, if after the bisection, one of the resulted two clusters contains less than 5% of all the documents, we consider that the cluster is not separable according to the criterion function. In such cases, we keep the cluster as it is and do not select it for further bisections. Thus, the number of clusters returned by our algorithm could be smaller than the number of required clusters as the input (if all the resulted clusters meet the stop condition).

6.2.2 Computational Complexity

Each iteration of the refinement of a 2-way clustering of a set of documents requires the examination of the movement of each one of the documents to the other cluster. During this process, the most expensive computation is the calculation of the membership values which need to be updated for all the documents. Thus, the time complexity of each iteration is $O(n^2)$. If we assume that each successive bisection splits the documents into two roughly equal-size clusters and that we follow a larger-cluster selection strategy, then the overall amount of time required to compute all $k-1$ bisections is $O(n^2)$.

7. EXPERIMENTAL RESULTS

We experimentally evaluated the performance of the various soft criterion functions and compared them with the corresponding hard criterion functions using a number of different datasets. In the rest of this section we first describe the various datasets and our experimental methodology, followed by a description of the experimental results.

7.1 Document Collections

In our experiments, we used a total of twelve different datasets, whose general characteristics are summarized in Table 1. The smallest of these datasets contained 356 documents and the largest contained 1,170 documents. To ensure diversity in the datasets, we obtained them from different sources. For all datasets, we used a stop-list to remove common words, and the words were stemmed using Porter's suffix-stripping algorithm [33]. Moreover, any term that occurs in fewer than two documents was eliminated.

The *hitech* and *sports* datasets were derived from the San Jose Mercury newspaper articles that are distributed as part of the TREC collection (TIPSTER Vol. 3). Each one of these datasets were constructed by selecting documents that are part of certain topics in which the various articles were categorized (based on the *DESCRIPT* tag). The *hitech* dataset contained documents about computers, electronics, health, medical, research, and technology; and the *sports* dataset contained documents about baseball, basketball, bicycling, boxing, football, golfing, and hockey. The datasets *k1a*, *k1b*, and *wap* are from the WebACE project [31, 18, 5, 6]. Each document corresponds to a web page listed in the subject hierarchy of Yahoo! [40]. The datasets *k1a* and *k1b* contain exactly the same set of documents but they differ in how the documents were assigned to different classes. In particular, *k1a* contains a finer-grain categorization than that contained in *k1b*. The *fbis* dataset is from the Foreign Broadcast Information Service data of TREC-5 [39], and the classes correspond to the categorization used in that collection. The *la1* and *la2* datasets were obtained from articles of the Los Angeles Times that was used in TREC-5 [39]. The categories correspond to the *desk* of the paper that each article appeared

Table 1: Summary of data sets used to evaluate the various clustering criterion functions.

Data	Source	# of documents	# of terms	# of classes
hitech	San Jose Mercury (TREC)	767	7499	6
sports	San Jose Mercury (TREC)	858	7163	7
reuters1	Reuters-21578	908	10582	3
odp	Open Directory Project	356	551	3
inspec1	Scientific Database	920	11803	3
wap	WebACE	780	7131	20
k1a	WebACE	1170	9527	20
k1b	WebACE	1170	9781	6
fbis	FBIS (TREC)	821	1997	17
la1	LA Times (TREC)	801	8449	6
la2	LA Times (TREC)	769	8333	6
re1	Reuters-21578	829	3221	25

and include documents from the entertainment, financial, foreign, metro, national, and sports desks. The dataset *re0* is from Reuters-21578 text categorization test collection Distribution 1.0 [27]. The *reuters1*, *odp*, and *inspec1* datasets are the datasets used by [29]. Refer to [29] for detailed information about these datasets. As a summary, *reuters1* was derived from the Reuters-21578 collection. The *reuters1* dataset contains documents from three classes: trade, acq and earn. The *odp* dataset was derived from the open directory project and consists of three classes: drugs, health, and sport. Finally, the *inspec1* dataset was derived from a scientific database and the documents are on the topics of back-propagation, fuzzy control, and pattern classification. Note that all the datasets used in our study do not contain documents with multiple class labels. The original *odp* and *inspec1* datasets in [29] contain some documents with multiple class labels. For the purpose of our study, we only selected those documents that only belong to one class.

7.2 Experimental Methodology and Metrics

For each one of the different datasets we obtained a 10-way and 20-way clustering solution that optimized the various hard and soft clustering criterion functions (Equations 1- 8). Specifically, for each hard criterion function, we compared it with the corresponding soft criterion functions with a fuzzy factor m of different values. The quality of a clustering solution was evaluated using the ***entropy*** measure that is based on how the various classes of documents are distributed within each cluster.

Given a particular cluster S_r of size n_r, the entropy of this cluster is defined to be

$$E(S_r) = -\frac{1}{\log q} \sum_{i=1}^{q} \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r},$$

where q is the number of classes in the dataset and n_r^i is the number of documents of the ith class that were assigned to the rth cluster. The entropy of the entire solution is defined to be the sum of the individual cluster entropies weighted according to the cluster size, *i.e.*,

$$\textit{Entropy} = \sum_{r=1}^{k} \frac{n_r}{n} E(S_r).$$

A perfect clustering solution will be the one that leads to clusters that contain documents from only a single class, in which case the entropy will be zero. In general, the smaller the entropy values, the better the clustering solution is.

To eliminate any instances that a particular clustering solution for a particular criterion function got trapped into a bad local optimum, in all of our experiments we found ten different clustering solutions. As discussed in Section 6.2 each of these ten clustering solutions correspond to the best solution (in terms of the respective criterion function) out of ten different initial partitioning and refinement phases.

7.3 Comparison of the Hard and Soft Criterion Functions

Our experiments were focused on evaluating the quality of the clustering solutions produced by the various hard and soft criterion functions when they were used to compute a k-way clustering solution via repeated bisections. The clustering solutions of various hard criterion functions were obtained by using CLUTO [23]. The results for the various datasets and criterion functions for 10-, and 20-way clustering solutions are shown in Tables 2 and 3, respectively.

The results for each dataset are shown in each subtable, in which each column corresponds to one of the four criterion functions. The results of the soft criterion functions with various fuzzy factor values are shown in the first five rows (labeled by the fuzzy factor values), and those of the various hard criterion functions are shown in the last row. The entries that are boldfaced correspond to the best values obtained for each column, (*i.e.*, for each criterion function, the best value among the hard and various soft criterion functions with different m values for each dataset), whereas the underlined entries correspond to the best values obtained among all the criterion functions for each dataset.

A number of observations can be made by analyzing the results in Table 2. First, for most datasets, introducing softness into each one of the four criterion functions improved the quality of the clustering solutions, and different trends can be observed in the relative improvement for different criterion functions. The $\mathcal{SI}_1$ criterion function outperformed $\mathcal{I}_1$ on eight datasets, among which the relative improvements were greater than 10% for six datasets with the largest improvement of 23%. The effect of introducing softness was less significant, but more consistent for both $\mathcal{SI}_2$ and $\mathcal{SG}_1$ than that for $\mathcal{SI}_1$. For only three datasets, $\mathcal{SI}_2$ and $\mathcal{SG}_1$ performed better than $\mathcal{I}_2$ and $\mathcal{G}_1$, respectively, by more than 10%. However, $\mathcal{SI}_2$ and $\mathcal{SG}_1$ outperformed $\mathcal{I}_2$ and $\mathcal{G}_1$ on ten and nine datasets, respectively. $\mathcal{SE}_1$ benefits the least with improvements observed on seven datasets and improvements greater than 10% observed on two datasets. Second, the fuzzy factor values that achieved the best clustering solutions seemed to vary for different datasets, which suggests that the proper fuzzy factor values may relate to some characteristics of the datasets and their class conformations. Finally, $\mathcal{SG}_1$ was less sensitive to the choice of fuzzy factor values

Table 2: The Entropies of the clustering solutions obtained by hard and soft criterion functions with various fuzzy factors.

	hitech 10-way				sports 10-way				reuters1 10-way			
Method	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$
$m=1$	0.757	0.644	0.674	**0.584**	0.431	0.245	0.208	0.200	0.228	0.229	0.287	0.283
$m=2$	0.627	0.639	0.618	0.596	0.497	0.248	0.219	0.161	0.205	**0.194**	0.232	0.222
$m=4$	0.616	0.612	0.615	0.599	0.270	0.250	0.119	0.170	**0.194**	0.210	**0.226**	**0.201**
$m=6$	**0.611**	**0.586**	0.586	0.603	0.274	**0.177**	**0.106**	**0.156**	0.264	0.231	0.298	0.214
$m=8$	0.618	0.594	0.582	0.587	0.294	0.204	0.133	0.161	0.359	0.296	0.370	0.255
hard	0.644	0.610	**0.573**	0.585	**0.252**	0.181	0.158	0.185	0.250	0.218	0.262	0.248

	odp 10-way				Inspec1 10-way				wap 10-way			
Method	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$
$m=1$	**0.236**	**0.210**	**0.216**	**0.224**	**0.326**	0.324	0.330	0.298	0.586	0.540	0.513	0.386
$m=2$	0.277	0.246	0.218	0.247	0.365	0.303	0.297	0.303	0.570	0.412	0.411	0.387
$m=4$	0.326	0.289	0.309	0.282	0.469	0.297	0.300	0.295	0.595	**0.385**	0.399	0.376
$m=6$	0.379	0.327	0.335	0.308	0.390	0.306	0.286	0.300	0.456	0.415	**0.398**	**0.371**
$m=8$	0.421	0.308	0.355	0.346	0.400	0.302	0.320	0.297	0.454	0.443	0.412	0.409
hard	0.293	0.283	0.233	0.256	0.441	**0.293**	**0.283**	**0.290**	**0.421**	0.414	0.408	0.381

	k1a 10-way				k1b 10-way				fbis 10-way			
Method	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$
$m=1$	0.601	0.585	0.530	0.432	0.240	0.223	0.200	0.160	0.525	0.527	0.510	0.416
$m=2$	0.573	0.519	0.437	**0.413**	0.347	0.205	0.181	**0.112**	0.529	0.429	0.424	0.387
$m=4$	0.584	**0.418**	0.407	0.443	0.259	**0.153**	**0.125**	0.157	**0.375**	0.398	0.404	**0.372**
$m=6$	**0.448**	0.429	**0.406**	0.418	0.238	0.174	0.148	0.113	0.379	**0.380**	**0.394**	0.378
$m=8$	0.487	0.462	0.436	0.435	0.249	0.226	0.194	0.189	0.399	0.387	0.430	0.393
hard	0.460	0.434	0.410	0.419	**0.169**	0.154	0.153	0.133	0.396	0.396	0.398	0.382

	la1 10-way				la2 10-way				re1 10-way			
Method	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$
$m=1$	0.784	0.747	0.714	0.486	0.832	0.808	0.788	0.408	0.462	0.500	0.489	0.422
$m=2$	0.756	0.559	0.463	0.428	0.841	0.401	0.393	0.377	0.473	0.451	0.417	0.417
$m=4$	0.687	0.459	0.456	0.419	**0.381**	0.388	0.363	0.380	0.424	**0.386**	**0.394**	0.415
$m=6$	**0.459**	0.423	0.469	0.422	0.408	**0.358**	0.380	0.369	0.424	0.409	0.396	**0.397**
$m=8$	0.465	0.448	0.467	0.421	0.357	0.374	0.418	0.417	0.442	0.393	0.405	0.412
hard	0.519	**0.423**	**0.413**	**0.418**	0.457	0.400	**0.338**	**0.367**	**0.414**	0.397	0.396	0.404

than the other three criterion functions. Similar trends can be observed from Table 3 as well.

Note that for the *fbis* and *re1* datasets, the results of 10- and 20-way clustering solutions are the same for the $\mathcal{I}_1$ and various $\mathcal{SI}_1$ criterion functions. That is because we employed a strategy to stop bisecting a cluster as described in Section 6.2.1. The actual number of clusters obtained by the $\mathcal{I}_1$ and various $\mathcal{SI}_1$ criterion functions is smaller than ten for both *fbis* and *re1*.

8. DISCUSSION

The experimental results presented in the previous section suggest that for most datasets soft criterion functions can improve the quality of the clustering solutions. In this section, we look at the clustering solutions obtained by soft criterion functions more carefully and identify some of the different characteristics observed in clustering solutions obtained by hard and soft criterion functions.

8.1 The effect of fuzzy factor

We first look at how the fuzzy factor effects the moves made by the various soft criterion functions. Recall that the fuzzy factor controls the "softness" of the membership function and hence the "softness" of the clustering solutions. As m increases, the soft membership function becomes the hard membership function (*i.e.*, $\mu_{ij} = 1$, if d_i is most close to S_j; $\mu_{ij} = 0$, otherwise), and consequently soft criterion functions become hard criterion functions. Hence, for every move made by soft criterion functions, if we also compute the gain of the corresponding hard criterion function, we would expect that the agreement between soft and hard criterion functions will increase as m increases.

Figure 1(a) shows the average percentages of moves that were made by the various soft criterion functions and agreed by the corresponding hard criterion function. The percentage values were averaged over all the datasets. As shown in Figure 1(a), as m increases, we do see a trend of increasing agreement between the soft and the corresponding hard criterion functions for $\mathcal{SI}_1$, $\mathcal{SI}_2$, and $\mathcal{SE}_1$.

One of the interesting observations is that even though the degree to which the hard and soft criterion functions agree increases with increasing m, it does not reach very high values (*i.e.*, it does not approach 100%). This is true even for large values of m (not shown in the graph). The reason for that is that the hard clustering solution induced by the soft clustering algorithm will assign each document to the cluster for which it has the highest membership function. However, this cluster may not necessarily be the one that optimizes

Table 3: The Entropies of the clustering solutions obtained by hard and soft criterion functions with various fuzzy factors.

	hitech 20-way				sports 20-way				reuters1 20-way			
Method	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$
$m=1$	0.746	0.591	0.639	**0.513**	0.187	0.206	0.165	0.150	0.218	0.211	0.251	0.233
$m=2$	0.598	0.586	0.554	0.550	0.175	0.194	0.181	0.131	0.202	0.189	0.215	0.183
$m=4$	0.587	0.551	0.566	0.544	0.170	0.148	0.108	**0.112**	**0.180**	0.202	0.211	0.185
$m=6$	**0.578**	**0.523**	0.545	0.550	**0.164**	**0.128**	**0.099**	0.133	0.233	0.200	0.263	**0.182**
$m=8$	0.583	0.532	0.529	0.550	0.221	0.144	0.125	0.109	0.283	0.259	0.313	0.218
hard	0.592	0.553	**0.524**	0.541	0.177	0.138	0.122	0.141	0.198	**0.159**	**0.206**	0.186

	odp 20-way				Inspec1 20-way				wap 20-way			
Method	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$
$m=1$	**0.227**	0.153	0.176	**0.208**	**0.326**	0.324	0.330	0.282	0.578	0.523	0.494	0.324
$m=2$	0.234	**0.151**	**0.150**	0.220	0.363	0.301	0.290	0.290	0.559	0.321	0.309	0.307
$m=4$	0.295	0.202	0.245	0.219	0.469	0.286	0.276	0.281	0.585	**0.308**	**0.308**	0.283
$m=6$	0.277	0.255	0.257	0.253	0.374	0.291	0.273	0.281	0.377	0.311	0.321	**0.277**
$m=8$	0.363	0.267	0.268	0.297	0.376	0.290	0.309	0.284	0.384	0.335	0.325	0.317
hard	0.281	0.228	0.201	0.217	0.427	**0.283**	**0.270**	**0.275**	**0.326**	0.329	0.319	0.307

	k1a 20-way				k1b 20-way				fbis 20-way			
Method	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$
$m=1$	0.596	0.579	0.525	0.363	0.226	0.200	0.187	0.125	0.525	0.504	0.469	0.343
$m=2$	0.570	0.495	0.350	0.343	0.344	0.181	0.123	0.096	0.529	0.374	0.362	0.311
$m=4$	0.574	**0.329**	**0.321**	0.340	0.256	0.104	0.107	0.120	**0.375**	0.321	0.330	**0.290**
$m=6$	**0.340**	0.339	0.333	0.344	**0.139**	0.110	0.125	0.099	0.379	**0.309**	**0.326**	0.304
$m=8$	0.414	0.364	0.345	0.361	0.152	0.157	0.154	0.144	0.399	0.319	0.357	0.319
hard	0.376	0.347	0.349	**0.334**	0.155	**0.076**	**0.105**	**0.091**	0.396	0.322	0.329	0.316

	la1 20-way				la2 20-way				re1 20-way			
Method	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{G}_1$
$m=1$	0.783	0.746	0.702	0.426	0.832	0.808	0.788	0.364	0.462	0.434	0.414	0.321
$m=2$	0.747	0.543	0.416	0.380	0.841	0.359	0.348	0.326	0.473	0.379	0.309	0.310
$m=4$	0.671	0.414	0.390	**0.379**	**0.339**	0.341	0.326	0.334	0.424	**0.280**	0.307	**0.277**
$m=6$	0.410	**0.382**	0.402	0.391	0.353	**0.313**	0.329	**0.319**	0.424	0.299	0.307	0.284
$m=8$	**0.401**	0.416	0.404	0.383	0.335	0.333	0.374	0.355	0.442	0.317	0.326	0.317
hard	0.447	0.385	**0.379**	0.386	0.390	0.333	**0.307**	0.334	**0.414**	0.299	**0.287**	0.300

the respective hard criterion function. Despite this fact, the trend, that the agreement between soft and the corresponding hard criterion functions increases as m increases, is still valid as shown in Figure 1(a).

For $\mathcal{SG}_1$, the agreement between $\mathcal{SG}_1$ and $\mathcal{G}_1$ seems less sensitive to the increasing of fuzzy factor values, which is consistent with the observation for $\mathcal{SG}_1$ from Tables 2 and 3.

8.2 Soft criterion functions tend to make moves more consistent with cluster separations.

We also looked at the degree to which the movement of documents between clusters, as being performed during the hard and soft criterion function optimizations, affects the inter-cluster separation. Specifically, every time a document is moved between two clusters (because such a move improves the overall value of the criterion function), we computed the cosine similarity between the cluster centroids before and after the move. Figure 1(b) shows the average percentages of the moves that also further separate the cluster centers for various criterion functions (*i.e.*, the cosine value between the two centroids decreased after the move). Again the percentages were averaged over all the datasets. The last data point for each criterion function represents the average percentage for the hard criterion function. As shown in Figure 1(b), the move made by hard criterion functions will not always increase the separation between cluster centers, whereas the soft criterion functions tend to make moves that are more consistent with cluster separations. For $\mathcal{SI}_1$, $\mathcal{SI}_2$ and $\mathcal{SE}_1$, more than 99.4% of the moves will separate cluster centers further when $m = 1$, and the percentage decreases as m increases (*i.e.*, the "softness" of clustering decreases). This property was also observed for fuzzy C-means [14] as well.

8.3 Soft criterion functions tend to lead to less balanced clustering solutions.

Another notable difference of clustering solutions obtained by hard and soft criterion functions is that soft criterion functions tend to lead to less balanced clustering solutions, and the smaller the fuzzy factor value is, the less balanced the clustering solution will be obtained. Table 4 gives an example of 10-way clustering solutions obtained by $\mathcal{SI}_2$ and $\mathcal{I}_2$ for *reuters1*.

The reason that soft criterion functions tend to lead to less balanced solutions is that since now one document can contribute to both clusters, the difference of soft sizes between two clusters will be smaller than that of hard sizes. Hence, soft criterion functions will tolerate clusters with higher difference in cluster sizes. Previous

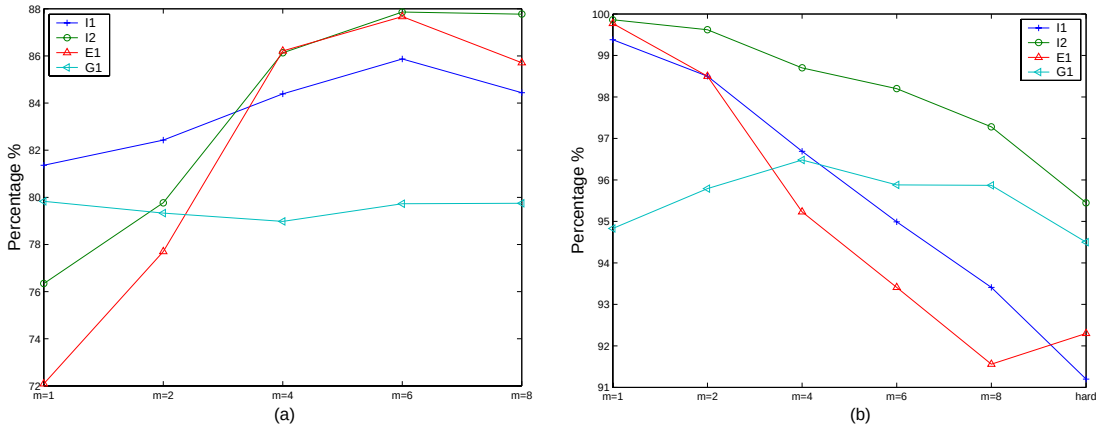

Figure 1: (a) Average percentages of the moves that made by both hard and soft criterion functions. (b) Average percentages of the moves that increase cluster separations.

Table 4: Comparison of 10-way clustering solutions obtained by $\mathcal{I}_2$ and $\mathcal{SI}_2$ for *reuters1*.

$\mathcal{SI}_2$ with $m = 2$ (Entropy=0.194)						$\mathcal{I}_2$ Criterion (Entropy=0.218)					
cid	Size	Sim	trad	acq	earn	cid	Size	Sim	trad	acq	earn
0	58	+0.220	0	0	58	0	83	+0.185	24	48	11
1	119	+0.220	0	0	119	1	67	+0.187	0	1	6
2	34	+0.219	32	0	2	2	136	+0.186	0	2	134
3	82	+0.188	22	48	12	3	76	+0.153	74	2	0
4	62	+0.171	59	3	0	4	67	+0.118	66	0	1
5	45	+0.090	0	5	40	5	88	+0.096	86	2	0
6	139	+0.078	137	2	0	6	79	+0.076	0	75	4
7	51	+0.074	0	46	5	7	85	+0.067	0	83	2
8	152	+0.054	0	146	6	8	98	+0.049	0	71	27
9	166	+0.035	1	160	5	9	129	+0.038	1	126	2

studies [46, 13] showed that highly unbalanced clusters will harm the quality of clustering solutions, hence the proper fuzzy factor should not be too small. Note that from the discussion in Section 8.2, we know that as the value of the fuzzy factor increases, a large fraction of the moves will not lead to better cluster separations. Hence, the fuzzy factor value that lead to the best clustering solution has to achieve the balance between these two factors.

9. CONCLUSION AND FUTURE RESEARCH

In this paper we extended four criterion functions that were studied in our previous work [46] to tackle the soft document clustering problem. We developed an approach similar to the induced fuzzy partition [1] to optimize various soft criterion functions. We presented a comprehensive experimental evaluation involving twelve different datasets and some discussions about the various trends observed from experimental results. Our experimental results and analysis show that the soft criterion functions tend to consistently improve the separation between the clusters, and lead to better clustering results for most datasets.

We plan to extend the work in this paper along three directions. First, develop and evaluate soft clustering extensions for the remaining three criterion fuctnions studied in [46], which includes some of the schemes that optimize internal and external characteristics of clustering solutions. Second, expand our evaluation to determine the effectiveness of these soft criterion functions to produce overlapping clustering solutions. Third, further understand $\mathcal{SG}_1$ and the reason why it is less sensitive to the value of the fuzzy factor as discussed in Section 8.

10. REFERENCES

[1] E. Backer. *Cluster Analysis by Optimal Decomposition of Induced Fuzzy Sets*. Delft University Press, Delft, The Netherlands, 1978.

[2] Doug Beeferman and Adam Berger. Agglomerative clustering of a search engine query log. In *Proc. of the Sixth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pages 407–416, 2000.

[3] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[4] L. Bobrowski and J.C. Bezdek. C-means clustering with the l1 and l norms. *IEEE Transactions on Systems, Man, Cybernetics*, 21(3):545–554, 1991.

[5] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Document categorization and query generation on the world wide web using WebACE. *AI Review)*, 11:365–391, 1999.

[6] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Partitioning-based clustering for web document categorization. *Decision Support Systems (accepted for publication)*, 1999.

[7] Daniel Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4), 1998.

[8] P. Cheeseman and J. Stutz. Baysian classification (autoclass): Theory and results. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI/MIT Press, 1996.

[9] D.R. Cutting, J.O. Pedersen, D.R. Karger, and J.W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the ACM SIGIR*, pages pages 318–329, Copenhagen, 1992.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39, 1977.

[11] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Knowledge Discovery and Data Mining*, pages 269–274, 2001.

[12] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1/2):143–175, 2001.

[13] Chris Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst Simon. Spectral min-max cut for graph partitioning and data clustering. Technical Report TR-2001-XX, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, 2001.

[14] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[15] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: An efficient clustering algorithm for large databases. In *Proc. of 1998 ACM-SIGMOD Int. Conf. on Management of Data*, 1998.

[16] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK: a robust clustering algorithm for categorical attributes. In *Proc. of the 15th Int'l Conf. on Data Eng.*, 1999.

[17] Greg Hamerly and Charles Elkan. Alternatives to the k-means algorithm that find better clusterings. In Konstantinos Kalpakis, Nazli Goharian, and David Grossmann, editors, *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM-02)*, pages 600–607, New York, November 4–9 2002. ACM Press.

[18] E.H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. WebACE: A web agent for document categorization and exploartion. In *Proc. of the 2nd International Conference on Autonomous Agents*, May 1998.

[19] E.H. Han, G. Karypis, V. Kumar, and B. Mobasher. Hypergraph based clustering in high-dimensional data sets: A summary of results. *Bulletin of the Technical Committee on Data Engineering*, 21(1), 1998.

[20] A.K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[21] A. Joshi and R. Krishnapuram. Robust fuzzy clustering methods to support web mining. In *Proc. of SIGMOD'98 Workshop on Data Mining and knowledge Discovery*, pages 15/1–15/8, 1998.

[22] G. Karypis, E.H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, 32(8):68–75, 1999.

[23] George Karypis. CLUTO a clustering toolkit. Technical Report 02-017, Dept. of Computer Science, University of Minnesota, 2002. Available at http://www.cs.umn.edu˜cluto.

[24] B. King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 69:86–101, 1967.

[25] D. H. Kraft, J. Chen, and A. Mikulcic. Combining fuzzy clustering and fuzzy inference in information retrieval. In *Proc. of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2000*, pages 375–380, May 2000.

[26] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proc. of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1999.

[27] D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. *http://www.research.att.com/∼lewis*, 1999.

[28] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Symp. Math. Statist, Prob.*, pages 281–297, 1967.

[29] M. E. S. Mendes and L. Sacks. Evaluating fuzzy clustering for relevance-based information access. In *Proc. of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2003*, pages 648–653, May 2003.

[30] S. Miyamoto. Fuzzy multisets and fuzzy clustering of documents. In *Proc. of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2003*, pages 1191–1194, Dec. 2001.

[31] J. Moore, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher. Web page categorization and feature selection using association rule and principal component clustering. In *7th Workshop on Information Technologies and Systems*, Dec. 1997.

[32] R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. of the 20th VLDB Conference*, pages 144–155, Santiago, Chile, 1994.

[33] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[34] Jan Puzicha, Thomas Hofmann, and Joachim M. Buhmann. A theory of proximity based clustering: Structure detection by optimization. *PATREC: Pattern Recognition, Pergamon Press*, 33:617–634, 2000.

[35] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[36] P. H. Sneath and R. R. Sokal. *Numerical Taxonomy*. Freeman, London, UK, 1973.

[37] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[38] A. Strehl and J. Ghosh. Scalable approach to balanced, high-dimensional clustering of market-baskets. In *Proceedings of HiPC*, 2000.

[39] TREC. Text REtrieval conference. *http://trec.nist.gov*, 1999.

[40] Yahoo! Yahoo! http://www.yahoo.com.

[41] K. Zahn. Graph-tehoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, (C-20):68–86, 1971.

[42] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Bipartite graph partitioning and data clustering. In *CIKM*, 2001.

[43] B. Zhang, M. Hsu, and U. Dayal. K-harmonic means a data clustering algorithm, 1999.

[44] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proc. of Int'l. Conf. on Information and Knowledge Management*, pages 515–524, 2002.

[45] Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. Technical Report TR #01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN, 2001. Available on the WWW at *http://cs.umn.edu/˜karypis/publications*.

[46] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.